

[Advanced Search](#) [Preferences](#) [Language Tools](#) [Search Tips](#)

"continuous compilation" (recompilation OR recompile)

[Web](#) - [Images](#) - [Groups](#) - [Directory](#) - [News](#)

Searched the web for "continuous compilation" (recompilation OR recompile). Results 11 - 18 of about 20.5

**[PS] [A Framework for Remote Dynamic Program Optimization](#)**

File Format: Adobe PostScript - [View as Text](#)

... Plezbert and Cytron [13] have proposed **continuous compilation** to overlap the "just-in ... that are passed to a remote optimizer for optimization and **recompilation**. ...

[www.ece.purdue.edu/~eigenman/reports/dynamo.ps](http://www.ece.purdue.edu/~eigenman/reports/dynamo.ps) - [Similar pages](#)

**[PDF] [Practicing JUDO: Java Under Dynamic Optimizations](#)**

File Format: PDF/Adobe Acrobat

... the window for triggering **recompilation** is wider than the interpreting approach, allowing more flexibility in terms of determining when to **recompile**. ...

[portal.acm.org/ft\\_gateway.cfm?id=349306&type=pdf&dl=GUIDE&dl=ACM&CFID=11111111&CFTOKEN=...](http://portal.acm.org/ft_gateway.cfm?id=349306&type=pdf&dl=GUIDE&dl=ACM&CFID=11111111&CFTOKEN=...) -

[Similar pages](#)

**[PS] [A Brief History of Just-In-Time \(Preprint\) John Aycock Department ...](#)**

File Format: Adobe PostScript - [View as Text](#)

... This dynamic **recompilation** process could be repeated any number of times ... and Cytron [1997] proposed and evaluated the idea of "**continuous compilation**" for Java ...

[codespeak.net/svn/pypy/trunk/doc/funding/jit-history.ps](http://codespeak.net/svn/pypy/trunk/doc/funding/jit-history.ps) - [Similar pages](#)

**[PS] [Short Title: Continuous Compilation Plezbert, M.Sc. 1996](#)**

File Format: Adobe PostScript - [View as Text](#)

... **Continuous Compilation**. ... compiler: to determine if some files must be compiled before others and to determine which files need **recompilation** after another file ...

[www.cs.wustl.edu/~plezbert/contcom/thesis-main.ps](http://www.cs.wustl.edu/~plezbert/contcom/thesis-main.ps) - [Similar pages](#)

**[Distributed Make System](#)**

... project had the experience when he wasted a lot of time spent on building his project - the process involving **continuous compilation** and **recompilation** of the ...

[tochna.technion.ac.il/project/dMake/html/dMake.html](http://tochna.technion.ac.il/project/dMake/html/dMake.html) - 9k - Supplemental Result - [Cached](#) - [Similar pages](#)

**[PS] [ONLINE PROFILING AND FEEDBACK-DIRECTED](#)**

File Format: Adobe PostScript - [View as Text](#)

... or **recompile** at one of Jikes RVM's three optimization-levels (O0, O1, O2). The controller estimates the cost and benefit of each potential **recompilation** choice ...

[www.cs.rutgers.edu/~marnold/papers/thesis.ps](http://www.cs.rutgers.edu/~marnold/papers/thesis.ps) - [Similar pages](#)

**[PS] [A SUIF Java Compiler](#)**

File Format: Adobe PostScript - [View as Text](#)

... Vortex implements traditional optimizations, selective **recompilation**, an iterative data flow analysis framework, and exceptions. ...

[www.cs.ucsb.edu/research/trcs/docs/1998-18.ps](http://www.cs.ucsb.edu/research/trcs/docs/1998-18.ps) - [Similar pages](#)

**[From ifeffit@millenia.cars.aps.anl.gov Mon Dec 2 15:19:46 2002 ...](#)**

... all steps > necessary, (it required ten hours of **continuous compilation**) for installing ... This will download and **recompile** all packages which have changed to the ...

[millenia.cars.aps.anl.gov/pipermail/ifeffit/2002-December.txt](mailto:millenia.cars.aps.anl.gov/pipermail/ifeffit/2002-December.txt) - 101k - Supplemental Result - [Cached](#) - [Similar](#)

[pages](#)

[> home](#) [> about](#) [> feedback](#) [> log](#)

US Patent &amp; Trademark Office

Try the *new* Portal design

Give us your opinion after using it.

Search DL [> Advanced Search](#) [> Search Help/](#)

## ACM Digital Library

A half century of pioneering concepts and fundamental research have been digitized and indexed in a variety of ways in this special collection of works published by ACM since its inception. The ACM Digital Library includes bibliographic information, abstracts, reviews, and full texts.

### Digital Library Overview

- [What's New](#)
- [FAQ](#)
- [DL Pearls](#)
- [Content and Organization](#)
- [Terms of Usage](#)
- [Resources from Affiliated Organizations](#)

### Subscription and Access Information

- > [Access Information](#)
- > [Individual Subscriptions](#)
- > [Institutional Subscriptions](#)

## Browse the Digital Library

- [Journals](#)
- [Magazines](#)
- [Transactions](#)
- [Proceedings](#)
- [Newsletters](#)
- [Publications by Affiliated Organizations](#)
- [Special Interest Groups \(SIGs\)](#)

## Personalized Services

- [My Bookshelf](#) Custom collections. Personal virtual Journals. Intelligent agent searches. Collaborative filtering.



## Online Computing Reviews Service

- [OCRS](#) Access critical reviews of the computing literature using the Online Computing Reviews Service.



[> home](#) [> about](#) [> feedback](#) [> login](#)

US Patent & Trademark Office



Try the *new* Portal design

Give us your opinion after using it.

## Search Results


Search Results for: **[continuous compilation]**  
Found **14** of **126,269** searched.

## Search within Results






[> Advanced Search](#)

[> Search Help/Tips](#)

Sort by: **Title** **Publication** **Publication Date** **Score**  Binder

**Results 1 - 14 of 14**    **short listing**

- 1** Overlapping execution with transfer using non-strict execution for 84%  
 mobile programs  
 Chandra Krintz , Brad Calder , Han Bok Lee , Benjamin G. Zorn  
**Proceedings of the eighth international conference on Architectural support for programming languages and operating systems** October 1998  
 Volume 32 , 33 Issue 5 , 11  
 In order to execute a program on a remote computer, it must first be transferred over a network. This transmission incurs the over-head of network latency before execution can begin. This latency can vary greatly depending upon the size of the program, where it is located (e.g., on a local network or across the Internet), and the bandwidth available to retrieve the program. Existing technologies, like Java, require that a file be fully transferred before it can start executing. For large files an ...
- 2** Proxy compilation of dynamically loaded Java classes with MoJo 82%  
 Matt Newsome , Des Watson  
**ACM SIGPLAN Notices , Proceedings of the joint conference on Languages, compilers and tools for embedded systems: software and compilers for embedded systems** June 2002  
 Volume 37 Issue 7  
 Interest in Java implementations for resource-constrained environments such as embedded systems has been tempered by concerns regarding its efficiency. Current native compilers for Java offer dramatic increases in efficiency, but have poor support for dynamically-loaded classes, which are typically served by slow interpreters or JIT compilers, the code-size of this latter utterly mismatching the resource constraints of the system. After a brief survey of Ahead-of-Time compilers for Java, we present ...
- 3** A brief history of just-in-time 80%  
 John Aycock  
**ACM Computing Surveys (CSUR)** June 2003  
 Volume 35 Issue 2  
 Software systems have been using "just-in-time" compilation (JIT) techniques since

the 1960s. Broadly, JIT compilation includes any translation performed dynamically, after a program has started execution. We examine the motivation behind JIT compilation and constraints imposed on JIT compilation systems, and present a classification scheme for such systems. This classification emerges as we survey forty years of JIT work, from 1960--2000.

#### 4 Geometric modelling in ALGOL 68

77%



I. C. Braid , R. C. Hillyard

**ACM SIGPLAN Notices , Proceedings of the Strathclyde ALGOL 68 conference**

March 1977

Volume 12 Issue 6

The paper describes the experiences of a small team in writing a substantial ALGOL 68 program to model the shapes of engineering components. The application derives much advantage from structures, operators and the heap. It includes a command interpreter, graphics package, vector and matrix routines, and procedures for moving data structures on the heap to and from disc. A system has been devised to ensure safe, selective compilation of program segments.

#### 5 Remote evaluation

77%



James W. Stamos , David K. Gifford

**ACM Transactions on Programming Languages and Systems (TOPLAS)** October 1990

Volume 12 Issue 4

A new technique for computer-to-computer communication is presented that can increase the performance of distributed systems. This technique, called remote evaluation, lets one computer send another computer a request in the form of a program. A computer that receives such a request executes the program in the request and returns the results to the sending computer. Remote evaluation provides a new degree of flexibility in the design of distributed systems. In present distributed systems th ...

#### 6 Third generation compiler design

77%



Marvin V. Zelkowitz

**Proceedings of the 1975 annual conference** January 1975

Compilers, besides testing for errors in a particular implementation of an algorithm, can be implemented to analyze program structure. This information can be fed back to the programmer in order to improve the structure, reliability and efficiency of the resulting program. This paper surveys several techniques that are currently implementable in a compiler, describes several new techniques that can be applied to programs, and briefly describes one such implementation of many of these ideas. ...

#### 7 High-level adaptive program optimization with ADAPT

77%










Michael J. Voss , Rudolf Eigemann

**ACM SIGPLAN Notices , Proceedings of the eighth ACM SIGPLAN symposium on Principles and practices of parallel programming** June 2001

Volume 36 Issue 7

Compile-time optimization is often limited by a lack of target machine and input data set knowledge. Without this information, compilers may be forced to make conservative assumptions to preserve correctness and to avoid performance degradation. In order to cope with this lack of information at compile-time, adaptive and dynamic systems can be used to perform optimization at runtime when complete knowledge of input and machine parameters is available. This paper presents a compiler-supporte ...

- 8** Practicing JUDO: Java under dynamic optimizations 77%  
 Michał Cierniak , Guei-Yuan Lueh , James M. Stichnoth  
**ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation** May 2000  
 Volume 35 Issue 5  
 A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient implementation of Just-In-Time (JIT) compilation, exception handling, synchronization mechanism, and garbage collection (GC). These components are tightly coupled to achieve high performance. In this paper, we present some static and dynamic techniques implemented in the JIT compilation and exception handling of the Microprocessor Research Lab Virtual Machine (MRL VM), ...
- 9** PL360, a Programming Language for the 360 Computers 77%  
 Niklaus Wirth  
**Journal of the ACM (JACM)** January 1968  
 Volume 15 Issue 1  
 A programming language for the IBM 360 computers and aspects of its implementation are described. The language, called PL360, provides the facilities of a symbolic machine language, but displays a structure defined by a recursive syntax. PL360 was designed to improve the readability of programs which must take into account specific characteristics and limitations of a particular computer. It represents an attempt to further the state of the art of programming by encouraging and even forcing ...
- 10** Dynamic binding of separately compiled objects under program control 77%  
 Rex E. Gantenbein , Douglas W. Jones  
**Proceedings of the 1986 ACM fourteenth annual conference on Computer science** February 1986
- 11** President's letter 77%  
 John R. White  
**Communications of the ACM** November 1990  
 Volume 33 Issue 11
- 12** Efficient support of parallel sparse computation for array intrinsic functions of Fortran 90 77%  
 Rong-Guey Chang , Tyng-Ruey Chuang , Jenq Kuen Lee  
**Proceedings of the 12th international conference on Supercomputing** July 1998
- 13** Does "just in time" = "better late than never"? 77%  
 Michael P. Plezbert , Ron K. Cytron  
**Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages** January 1997
- 14** Compiling lazy pattern matching 77%  
 Luc Maranget  
**ACM SIGPLAN Lisp Pointers , Proceedings of the 1992 ACM conference on LISP and functional programming** January 1992  
 Volume V Issue 1

---

**Results 1 - 14 of 14**    *short listing*

---

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.